

Wavelet Sparse Approximate Inverse Preconditioners

Tony Chan W.-P. Tang and W. L. Wan

RIACS Technical Report 96.18

November 1996

Submitted to *BIT*.

A large black triangle is located in the bottom right corner of the page, pointing upwards and to the left.

Wavelet Sparse Approximate Inverse Preconditioners

Tony Chan W.-P. Tang and W. L. Wan

The Research Institute for Advanced Computer Science is operated by Universities Space Research Association, The American City Building, Suite 212, Columbia, MD 21044 (410)730-2656

Work reported herein was supported in part by NASA under contract NAS 2-96027 between NASA and the Universities Space Research Association (USRA).

WAVELET SPARSE APPROXIMATE INVERSE PRECONDITIONERS

TONY F. CHAN ^{*}, W. P. TANG [†] AND W. L. WAN^{*}

Abstract. There is an increasing interest in using sparse approximate inverses as preconditioners for Krylov subspace iterative methods. Recent studies of Grote and Huckle [21] and Chow and Saad [11] also show that sparse approximate inverse preconditioner can be effective for a variety of matrices, e.g. Harwell-Boeing collections. Nonetheless a drawback is that it requires rapid decay of the inverse entries so that *sparse* approximate inverse is possible. However, for the class of matrices that come from elliptic PDE problems, this assumption may not necessarily hold. Our main idea is to look for a basis, other than the standard one, such that a sparse representation of the inverse is feasible. A crucial observation is that the kind of matrices we are interested in typically have a piecewise smooth inverse. We exploit this fact by applying wavelet techniques to construct a better sparse approximate inverse in the wavelet basis. We shall justify theoretically and numerically that our approach is effective for matrices with smooth inverse.

We emphasize that in this paper we have only presented the idea of wavelet approximate inverses and demonstrated its potential but have not yet developed a highly refined and efficient algorithm.

Key words. preconditioning, approximate inverses, sparse matrices, wavelets

AMS(MOS) subject classifications. 65F10, 65F35, 65F50, 65Y05, 65Y20

1. Introduction. Preconditioners are often used to accelerate the solution process of Krylov subspace iterative methods for solving linear systems:

$$Ax = b,$$

where A is large and sparse. If in addition, A is derived from an elliptic PDE problem, optimal preconditioners, in the sense that the condition number of the preconditioned systems is independent of the mesh size, such as multigrid and domain decomposition, have been proposed [24] [9] [29]. However, they are not readily applicable to general matrices. Incomplete LU factorization (ILU) preconditioners are often used instead. In fact, ILU is widely used as a preconditioner to solve both general and PDE problems for its robustness. Unfortunately, the parallelization is not straightforward.

Ideally, we would like to have a parallel preconditioner which is robust for both general and PDE problems. The recent interest of sparse approximate inverse may be because it is a potential candidate for such preconditioner. On one hand, it possesses a conceptually straightforward parallel implementation. Moreover, the application of the preconditioner is simply matrix-vector multiply instead of backsolve and it can be done easily in parallel. On the other hand, due to its algebraic nature, it is applicable to both general and PDE problems. Moreover, recent studies of Grote and Huckle [21] and Chow and Saad [11] show that it is robust for matrices in Harwell-Boeing collections. The main idea of sparse approximate inverse is described as follows.

Consider solving the right preconditioned linear system,

$$(1) \quad AMy = b, \quad x = My,$$

^{*} Dept. of Mathematics, Univ. of Calif. at Los Angeles, Los Angeles, CA 90095-1555. Email: chan@math.ucla.edu, wlwan@math.ucla.edu. Supported by grants from ONR: ONR-N00014-92-J-1890, and the Army Research Office: DAAL-03-91-C-0047 (Univ. of Tenn. subcontract ORA4466.04 Amendment 1 and 2). The first and the third author also acknowledges support from RIACS/NASA Ames NAS 2-13721 and Alfred P. Sloan Foundation as Doctoral Dissertation Fellows, respectively.

[†] Dept. of Computer Science, Univ. of Waterloo, Waterloo, Ontario N2L 3G1, Canada. Email: wptang@yoho.waterloo.edu. The work was supported by the Natural Sciences and Engineering Research Council of Canada, the Information Technology Research Centre, (which is funded by the Province of Ontario), and RIACS/NASA Ames NAS 2-13721.

where A is large, sparse matrix and M is a right preconditioner. We want to find a sparse matrix M so that $\|AM - I\|$ is small in some norm. This approach was first studied by Benson [2] and Benson and Frederickson [3]. More precisely, they minimize the residual matrix,

$$(2) \quad \min_M \|AM - I\|_F^2,$$

subject to some constraint on the number and position of the nonzero entries of M . The Frobenius norm is particularly useful for parallel implementation. Notice that

$$\|AM - I\|_F^2 = \sum_{j=1}^n \|Am_j - e_j\|_2^2,$$

where m_j and e_j are the j th column of M and I respectively. Thus solving (2) leads to solving n independent least squares problems,

$$(3) \quad \min_{m_j} \|Am_j - e_j\|_2, \quad j = 1, \dots, n,$$

which can be done in parallel.

Another possibility is to use a weighted Frobenius norm which had been investigated intensively by Kolotilina et al [26] [25] [27]. A complete survey can be found in [1]. Other constructions of approximate inverse are discussed in [11] [10] [30] [4] [5] [19] [8]. A comparison of approximate inverse preconditioners and ILU(0) on Harwell-Boeing matrices can be found in [20]. In this paper, however, we shall focus primarily on the Frobenius norm approach.

In practice, it is desirable to look for sparse solution of (3). However, this poses two difficulties: how to determine the sparsity pattern of M and how to solve (3) efficiently. Recently, two main approaches have been suggested. One is discussed by Cosgrove et al [12] and Grote and Huckle [21] and the other is by Chow and Saad [11]. For the former approach, they solve the least squares problems (3) by QR factorization, which may sound costly. But since m_j is sparse, the cost of QRF can be greatly reduced. Moreover, they derive algorithms to determine the positions of fill-in adaptively. Similar methods can be found in [26], [25], [27], [22], [23], in which case, the sparsity pattern of M is typically fixed as banded or the nonzeros of A .

For Chow and Saad's approach, they use standard iterative method (e.g. GMRES) to find an approximate solution to

$$Am_j = e_j,$$

and apply some dropping strategy to m_j to control the amount of fill-in. The idea is to let the Krylov subspace build up the sparsity pattern gradually and then the nonzeros entries are selected automatically by size. We should also mention that they have suggested several other possibilities of solving (2) [11].

A major drawback on the use of sparse approximate inverse preconditioning for elliptic PDEs is the assumption that A^{-1} can be approximated by a sparse matrix M . It is well known that A^{-1} will, in general, be dense even if A is sparse. Yet one might expect rapid decay in the A^{-1} entries away from the diagonal [15], [7], [16], [17] for some class of matrix A , e.g. banded or diagonal dominant matrix. However, as mentioned in [12], [11], if we require $\|AM - I\|_1 < 1$, we can always find a sparse matrix A while M has to be structurally dense. Even if A is symmetric positive definite or A comes from PDE problems, the inverse entries need not decay fast enough.

For example, consider the following near tridiagonal symmetric positive definite matrix of size 40×40 , derived from some artificial periodic boundary like problem,

$$(4) \quad A = \begin{bmatrix} 2.01 & -1 & & & -1 \\ -1 & 2.01 & -1 & & \\ & -1 & 2.01 & -1 & \\ & & \ddots & \ddots & \ddots \\ & & & -1 & 2.01 & -1 \\ -1 & & & & -1 & 2.01 \end{bmatrix}.$$

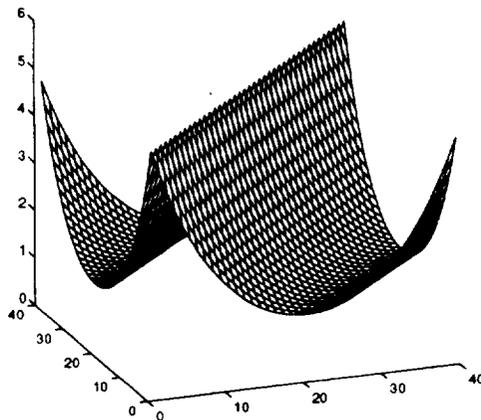


FIG. 1. Mesh plot of A^{-1} .

As we can see from Fig 1, the inverse entries are all greater than one and hence have no decay at all.

We also remark that even if A^{-1} has decay away from the diagonal (e.g. A comes from the Laplace operator), the rate of decay may not be enough for the approximate inverse to have optimal convergence, in the sense that the number of iteration for convergence is independent of mesh size. This is verified numerically in Table 1, where SPAI is the sparse approximate inverse given by Grote and Huckle's implementation [21]. The number in the bracket is the maximum allowable size of the residual norm of each column. In general, the smaller the number is, the better (but also the denser) the approximate inverse is.

h	no. of GMRES(20) iter			no. of nonzeros in precondition.		
	SPAI(0.4)	SPAI(0.2)	ILU(0)	SPAI(0.4)	SPAI(0.2)	ILU(0)
1/8	16	10	9	208	696	288
1/16	29	17	14	1040	3640	1216
1/32	67	37	25	4624	16440	4992
1/64	160	63	57	19472	69688	20224

TABLE 1
Convergence of GMRES(20) where $A=2D$ Laplacian.

For PDE problems, sufficient decay of inverse entries does not necessarily happen. More likely to occur, however, is piecewise smoothness of the entries in the rows and columns of the matrix. The key observation is that if A corresponds to a differential operator of some elliptic PDE, the inverse would be the corresponding discrete Green's function. (See appendix for a simple example illustrating this idea.) Similar observation in the case of solving integral equation can be found in [6] and in the case of hyperbolic and parabolic PDE can be found in [18]. Since A^{-1} corresponds to the Green's function, we would expect piecewise smooth changes in the inverse entries, with singularity along the diagonal. In other words, if we treat the inverse matrix as a graph of a function of two variables, then we will get a piecewise smooth graph. Note that piecewise smoothness is more general in the sense that the entries need not have decay as shown in Figure 1. From now on, we shall focus our attention on this kind of matrices and try to derive an efficient approximate inverse preconditioner.

Our main idea is to convert the smooth entries of A^{-1} into small numbers so that approximate inverse can be effective. The technique is to look for a basis such that A^{-1} has a sparse (modulo small numbers) representation. In fact the idea is similar to that of Beylkin et al [6] and Engquist et al [18]. Our strategy is to apply wavelet transforms to compress these piecewise smooth inverse matrices. Then we combine this idea with standard approximate inverse techniques (e.g. Grote and Huckle's implementation) to construct the sparse approximate inverse. The use of wavelets to solve integral and differential equations can also be found in [6], [18], [19], [8].

In section 2, we show how to adopt the wavelet transform in the least squares approach to solve (3). In section 3, we justify theoretically that a smooth inverse has better decay in the wavelet basis. We also make an interesting connection between our wavelet based preconditioner and the classical hierarchical basis preconditioner. In section 5, we estimate the extra cost for the wavelet transform and discuss the implementation issues of how to simplify the algorithm. In section 6, we present several numerical examples to compare the various methods. Finally, we make some conclusions in section 7.

We would like to remark that the purpose of this paper is not to present an ultimate algorithm for solving linear systems. Rather, we address a conceptual weakness of the standard approximate inverse technique and propose a way to remedy it. The main emphasis is on how to model the approximate inverse appropriately in order to solve a certain class of problems, e.g. matrices with piecewise smooth inverse. Many algorithmic variants are in fact possible but have not yet been fully explored. It is hoped that this paper will lead to further development and improvement of algorithms.

2. Fast Wavelet Based Approximate Inverse. The advantage of applying wavelet technique is to convert possibly large but smooth entries into small quantities so that the *sparse* approximate inverse can be more effective. Although Fourier basis also possesses similar approximation properties, we prefer wavelets to Fourier basis because the inverses we are approximating typically have local singularities which could cause the Gibbs phenomenon if Fourier transforms were used. Wavelet transforms, on the other hand, are local both in space and frequency. Intuitively speaking, for a function in L^2 with piecewise smoothness, only a small number of wavelet basis functions is needed to represent it well.

Similar to Fourier basis, wavelets can also be realized in a discrete sense. Given an orthogonal wavelet function in the continuous space, there corresponds an orthogonal matrix W that transforms vectors from the standard basis to the wavelet basis. Furthermore, if v is a vector of smoothly varying numbers (with possibly local singularities), its wavelet representation $\tilde{v} = Wv$, will have mostly small entries. We shall make use of this remarkable property to construct our sparse approximate inverse.

We can also represent two dimension transforms by W . Let A be a matrix in the standard basis. Then $\tilde{A} = WAW^T$ is the representation of A in the wavelet basis. This wavelet representation \tilde{A} is also called the standard form of A [6]. Nonstandard forms of A also exist but we do not discuss this further.

Since we are only interested in the application of wavelets to construct an approximate inverse, we only mention a few features of wavelets. See e.g. [14] for more detail description of wavelets.

Assuming A^{-1} is piecewise smooth, our idea is to apply wavelet transform to compress A^{-1} and then use it as a preconditioner. At first glance, this seems impossible since we do not even have A^{-1} . Our trick is the observation that

$$\widetilde{A^{-1}} \equiv WA^{-1}W^T = (WAW^T)^{-1} = \tilde{A}^{-1},$$

where W is an orthogonal wavelet transform matrix. Therefore we can first transform A to its wavelet basis representation \tilde{A} and then apply, for example, Grote and Huckle's method to find an approximate inverse for \tilde{A} , which is the preconditioner that we want to compute. In other words, we do not need to form A^{-1} but are still able to compute its transform. We shall make all these ideas more precise in the next section.

2.1. Wavelet Formulation. We shall show how we adopt wavelet transform in the least squares approach. Consider equation (2) again. Let W be an orthogonal wavelet transform matrix, i.e. $\tilde{x} = Wx$ is the vector x in the wavelet basis. (Note that W can be 1-level or full $\log_2 n$ -level wavelet transform matrix.) Then

$$\begin{aligned} (5) \quad \min_M \|AM - I\|_F &= \min_{\tilde{M}} \|WAW^T W M W^T - I\|_F \\ &= \min_{\tilde{M}} \|\tilde{A}\tilde{M} - I\|_F, \end{aligned}$$

where $\tilde{A} = WAW^T$ and $\tilde{M} = W M W^T$ are the representations of A and M in the wavelet basis respectively. Thus, our n least squares problems become

$$(6) \quad \min_{\tilde{m}_j} \|\tilde{A}\tilde{m}_j - \epsilon_j\|_2, \quad j = 1, 2, \dots, n.$$

\tilde{A} is sparse (but probably denser than A) since A is. Because of the wavelet basis representation, if M is piecewise smooth, we would expect \tilde{M} , neglecting small entries, to be sparse too. Therefore, the sparse solution of (6) would hopefully give rise to a more effective approximate inverse than the original approach without the wavelet transform. We shall justify our claim numerically in section 6.

2.2. Algorithm. In the previous section, we derive the wavelet formulation of our approximate inverse. Then we can simply apply the standard techniques of approximate inverse (e.g. Grote and Huckle's implementation, which is called the SPAI algorithm [21]), to solve the n least squares problems (6) in parallel. Here is the algorithm.

Wavelet Based Approx Inverse Algorithm

- (a) Wavelet transform A to get $\tilde{A} = WAW^T$.
- (b) Apply standard approximate inverse algorithm (e.g. SPAI) to solve for \tilde{M} .
- (c) Use \tilde{M} as preconditioner to solve: $\tilde{A}\tilde{x} = \tilde{b}$, where $\tilde{b} = Wb$.
- (d) Apply backward wavelet transform to \tilde{x} to obtain $x = W^T\tilde{x}$.

It should be noted that if we know the sparsity pattern of \tilde{M} *a priori*, we can simply use that pattern to solve the least squares problems (6) instead of using Grote and Huckle's adaptive approach. It will then be much more efficient. We shall discuss the implementation issue further in section 5.

3. Theoretical Aspect. Our wavelet based approximate inverse relies on the ability of wavelets to change (local) smoothness to small wavelet coefficients. In this section, we shall combine the classical result of Beylkin et al [6] and our construction to derive an residual estimate for our preconditioner.

In the discussion below, we shall follow the notation in [6]. We list some useful definitions which will be used later. Define the set of dyadic intervals on $[0,1]$ by,

$$\mathcal{I} = \{[\epsilon^{-l}, \epsilon^{-l}(\|\ + \infty)] : l \leq \|\leq \epsilon^l - \infty, l \leq \|\leq \log_{\epsilon} \setminus \}$$

Let $I_{jk} = [2^{-j}k, 2^{-j}(k+1)] \in \mathcal{I}$. Then $|I_{jk}| = \text{length of } I_{jk}$ is defined as: $2^{-j}(k+1) - 2^{-j}k = 2^{-j}$.

In order to bound the size of the elements of \tilde{A}^{-1} in the wavelet basis, we need the following smoothness assumptions on the Green's function $G(x, y)$:

$$(7) \quad |G(x, y)| \leq \frac{1}{|x - y|},$$

$$(8) \quad |\partial_x^m G(x, y)| + |\partial_y^m G(x, y)| \leq \frac{C_m}{|x - y|^{m+1}},$$

for some $m \geq 1$ and $C_m \geq 0$.

The following is a classical result of Beylkin et al [6] on the estimate of integral operator.

THEOREM 3.1. *Suppose the Green's function $G(x, y)$ satisfies the smoothness assumptions (7) and (8). Let \tilde{A}^{-1} be the discrete operator of $G(x, y)$ in the wavelet basis. Then the (k, l) th entry of \tilde{A}^{-1} is bounded by,*

$$(\tilde{A}^{-1})_{k,l} \leq C_m \left(\frac{|I_k|}{|I_l|} \right)^{\frac{1}{2}} \left(\frac{|I_k|}{d(I_k, I_l)} \right)^{m+1},$$

where $I_k, I_l \in \mathcal{I}$, $|I_k| \leq |I_l|$ and $d(I_k, I_l) = \text{distance between } I_k \text{ and } I_l$.

From the bound, we can see that the length and position of I_k and I_l determine the size of $(\tilde{A}^{-1})_{k,l}$. By the definition of dyadic intervals and as mentioned in [6], $d(I_k, I_l)$ is equal or close to 0 at $O(n \log_2 n)$ locations only. In other words, effectively \tilde{A}^{-1} only has $O(n \log_2 n)$ number of elements for large enough n .

With this result, we are able to estimate the quality of our approximate inverse. Let $\epsilon > 0$ be given. Define a sparsity pattern \mathcal{S} to be:

$$\mathcal{S} = \{(\|\, l) : (\tilde{A}^{-1})_{\|\, l} \geq \epsilon\}.$$

Due to Theorem 3.1, the number of elements in $\mathcal{S} = \mathcal{O}(\lfloor \log_\epsilon \rfloor)$. We have the following estimate.

THEOREM 3.2. *If we choose \mathcal{S} as our sparsity pattern, then*

$$(9) \quad \|AM - I\|_F \leq n\|A\|_F\epsilon.$$

Proof. We first define an intermediate matrix \tilde{N} which is essentially the truncation of \tilde{A}^{-1} by,

$$(\tilde{N})_{i,j} = \begin{cases} (\tilde{A}^{-1})_{i,j} & (i,j) \in \mathcal{S}, \\ 0 & \text{otherwise,} \end{cases}$$

and denote the j th column of \tilde{N} by \tilde{n}_j . The inequality (9) is a direct consequence of (5), (6), the definition of least squares solution and the definition of \tilde{N} and is derived as follows:

$$\begin{aligned} \|AM - I\|_F^2 &= \|\tilde{A}\tilde{M} - I\|_F^2 \\ &= \sum_{j=1}^n \|\tilde{A}\tilde{m}_j - e_j\|_2^2 \\ &\leq \sum_{j=1}^n \|\tilde{A}\tilde{n}_j - e_j\|_2^2 \\ &= \|\tilde{A}\tilde{N} - I\|_F^2 \\ &= \|\tilde{A}(\tilde{N} - \tilde{A}^{-1})\|_F^2 \\ &\leq \|A\|_F^2 \|\tilde{N} - \tilde{A}^{-1}\|_F^2 \\ &\leq n^2 \|A\|_F^2 \epsilon^2. \end{aligned}$$

□

Remark: Similar bound can also be found in [21]. Our estimate is practically more useful in the sense that the sparsity pattern of \tilde{M} is only $\mathcal{O}(n \log_2 n)$ while those in [21] does not guarantee this.

The result of Theorem 3.2 is for theoretical interest only. First of all, the sparsity pattern $calS$ is not known in general. Besides, $\mathcal{O}(n \log_2 n)$ elements for \tilde{M} may be still too dense for practical purposes. Furthermore, because of the special finger-like distribution of the nonzero elements of \tilde{M} given by \mathcal{S} , the amount of computation for solving the least squares problems may differ substantially from column to column. Thus in our implementation, we only choose a subset of \mathcal{S} which corresponds to those entries near the main diagonal. We find that the quality is still promising as will be shown in section 6. The implementation detail will be discussed more in section 5.

4. Connection to hierarchical basis preconditioner. Because of the hierarchical structure of wavelets, there is a natural connection between our wavelet approximate inverse and the hierarchical basis preconditioner [33] [28]. Our wavelet approximate inverse, denoted by M^{WAI} , can be considered as an approximation to the transformed A^{-1} . That is,

$$(10) \quad \begin{aligned} M^{WAI} &= W^T \tilde{M} W, \\ \tilde{M} &= \mathit{approx}(\tilde{A}^{-1}), \end{aligned}$$

where $\mathit{approx}(\tilde{A}^{-1})$ is an approximation of \tilde{A}^{-1} . In our case, it is given by the solution of the least squares problems (6). We can also express the approximate inverse, M^{HB} , given by the hierarchical basis preconditioner in a similar form [28].

$$(11) \quad \begin{aligned} M^{HB} &= S^T M S, \\ \hat{M} &= (\mathit{approx}(A))^{-1}, \end{aligned}$$

where S^T is the non-orthogonal transformation matrix from the hierarchical basis to the standard basis, $\hat{A} = SAS^T$, and $\mathit{approx}(A)$ is another approximation of A , e.g. coarse grid operator of A .

These two approximate inverses are similar in that both possess a hierarchical structure. In fact, the hierarchical basis can also be considered as a special kind of wavelet since it consists of a hierarchy of piecewise linear functions and they are precisely the "hat" functions in the wavelet terminology. On one hand, our wavelet approximate inverse is more general than the one by the hierarchical basis in the sense that one is allowed to use other kind of wavelets, in particular, the orthogonal wavelets with compact support by Daubechies [13]. On the other hand, the converse is also true in the sense that one could apply the hierarchical basis transform in a more general domain.

The main difference between the two approximate inverses is the way they approximate the original matrix A . For the approximate inverse given by the hierarchical basis in (11), we first approximate \tilde{A} , e.g. by a block diagonal matrix, and then compute the exact inverse of it. For our wavelet approximate inverse, we compute \tilde{A} exactly and then approximate its inverse by solving the least squares problems as discussed in section 2. Typically, the approximate inverse given by the hierarchical basis is block diagonal with zero bandwidth except for the coarsest block while the one by wavelet can have nonzeros anywhere. If we choose the same block diagonal for the nonzeros of \tilde{M} , it will reduce to the same form (but probably with different values on the entries) of \tilde{M} .

Connections of wavelets and hierarchical basis are also made in [31], [32].

5. Complexity and Implementation of Algorithm. The naive algorithm in section 2.2 needs quite an amount of overhead for doing the wavelet transformation. In this section, we will analyze each step of the algorithm and discuss some implementation issues of how to simplify and speed up the procedures. Meanwhile, we also analyze the sequential complexity of each step. We shall show that it is essentially $O(n)$, except *step(a)* which requires $O(kn)$ operations, where k depends on the number of levels of the wavelet transform.

In the following discussion, we assume that the wavelet used is orthogonal and of compact support, [13], [14]. Orthogonal wavelets are used so that the formulation developed in section 2 makes sense. However, one could also use non-orthogonal wavelets anyway. Compact support, on the other hand, is indispensable so that the wavelet transform is only $O(n)$ and \tilde{A} does not become dense.

Step (a). In general, to compute the wavelet transform of a vector requires $O(n)$ operations. Computing $\tilde{A} = WAW^T$ is equivalent to transforming the columns and then the rows of A (or vice versa). Thus it will cost $O(n^2)$ operations. However, since A is sparse, if we assume that there are only $O(1)$ nonzeros in each column and each row, the cost will be reduced to $O(kn)$ where k , ranging from 1 to $\log_2 n$, is the number of levels in the wavelet transform. In fact, in the parallel implementation, we do not need to form \tilde{A} explicitly in each processor. Notice that solving each least squares problems only need a few columns of \tilde{A} . We just form those columns and the cost will be reduced to $O(n)$.

Step (b). In each level of the transform, we will introduce a fixed amount of fill-in. Even though there are only $O(1)$ nonzeros in each column and row of A , there will be $O(k)$ nonzeros in each column and row of \tilde{A} . We could choose k so small that the number of nonzero introduced is acceptable. We may also reduce the cost significantly by taking advantage of the fact that for smooth coefficient problems, the Green's function typically has singularity only at a point. In other words, A^{-1} only has singularity along the diagonal. Hence it is reasonable to fix the sparsity pattern of \tilde{M} to be block diagonal. In fact, our current implementation already assumes block diagonal structure for \tilde{M} . This assumption saves an enormous amount of time used for searching for the next nonzero entries adaptively. By the way, adaptive searching procedure is usually less amenable to parallel implementation. We may further reduce the cost by adopting the concept of *local inverse* in [30]. More precisely, we use the nonzero indices, i , of \tilde{m}_j for the column and row indices of \tilde{A} and then we compute the QR factorization for the submatrix $\tilde{A}(i, i)$ when solving (6). This reduces the overall cost of step (b) to $O(n)$.

Step (c). When we solve $\tilde{A}\tilde{x} = \tilde{b}$ by some iterative method, we need to perform \tilde{A} times a vector. If we do it directly, the cost will be $O(kn)$ as \tilde{A} has $O(kn)$ nonzeros in each row. Note that

$$\tilde{A}v = W(A(W^T v)).$$

If we first backward transform v , apply A and then transform it back, the overall process will only be $O(n)$.

6. Numerical Results. In this section, we compare our preconditioner with those by Grote and Huckle's SPAI and ILU(0). We choose several matrices that come from different elliptic PDE.

The inverses of all these matrices are piecewise smooth and the singularities are cluster around the diagonal. For efficiency, instead of applying SPAI to solve for \tilde{M} adaptively in step (a) of our algorithm, we specify a block diagonal structure *a priori* for \tilde{M} and then solve (6) by the QRF as discussed in section 5. In all the tests, we use the compact support wavelet D_4 by I. Daubechies [13], [14]. We apply 6 levels of wavelet transform to matrices of order 1024 and 8 levels of transform to matrices of order 4096. Note that the number of levels is arbitrary. One could use different number in different situations.

We apply these preconditioners to GMRES(20). The initial guess was $x_0 = 0, r_0 = b$ and the stopping criterion was $\|r_n\|/\|r_0\| < 10^{-6}$. All the experiments were done in MATLAB in double precision.

Example 1: We use two simple 1D matrices to show the benefit from using wavelet transforms. The first one is a slightly modified artificial matrix in (4) where the diagonal entries are changed from 2.01 to 2.00001 and the size is 1024×1024 . The second matrix is the 1D Laplacian operator derived from the following:

$$\begin{aligned} u''(x) &= f(x), & \text{in } (0, 1), \\ u(0) &= 0, \quad u'(1) = 0. \end{aligned}$$

Neumann boundary condition at $x = 1$ is used so that there is no decay in the Green's function near the boundary.

The bandwidth is 0,0,5,5,5,5 for the 1st to 6th level of the block diagonal structure of \tilde{M} respectively.

Example 2: In this case, A is the 2D Laplacian operator with size 1024×1024 and 4096×4096 . For $n=1024$, we choose the bandwidth of \tilde{M} as before. For $n=4096$, we choose the bandwidth = 0,0,0,0,5,5,5,5 for the 1st to 8th level of the block diagonals respectively.

Example 3: We try to solve something more complicated than the Laplace equation but still having a piecewise smooth inverse. Consider the following PDE with variable coefficients,

$$((1+x^2)u_x)_x + u_{yy} + (\tan y)^2 u_y = -100x^2.$$

We solve the 32×32 and 64×64 grid cases. The bandwidth of the block diagonal of \tilde{M} is the same as before.

Example 4: In this case, A comes from a PDE of helical spring:

$$u_{xx} + u_{yy} + \frac{3}{5-y} u_x - 2G\lambda = 0,$$

where G and λ are some constants. Same setting as before.

Example 5: Finally, we show an example where our wavelet preconditioner does not work. The matrix A comes from a discontinuous coefficients PDE:

$$(a(x, y)u_x)_x + (b(x, y)u_y)_y + u_x + u_y = \sin(\pi xy),$$

where the coefficients $a(x, y)$ and $b(x, y)$ are defined as:

$$a(x, y) = b(x, y) = \begin{cases} 10^{-3} & (x, y) \in [0, 0.5] \times [0.5, 1] \\ 10^3 & (x, y) \in [0.5, 1] \times [0, 0.5] \\ 1 & \text{otherwise.} \end{cases}$$

The bandwidth is chosen to be 5,5,10,10,15,15 to make the number of nonzeros comparable to that of SPAI(0.2). Such modification is made so that sparsity is not a factor for the failure.

The convergence of GMRES(20) with different preconditioners in each example is shown in Figures 2-5 and is summarized in Table 2. In Example 1, we can see that SPAI(0.4) and SPAI(0.2) converge very slowly in this somewhat artificial but illustrating case. On the other hand, the wavelet based preconditioner converges much faster. This shows the advantage of wavelet transform in the case

where A^{-1} is smooth with singularity only along the diagonal. We do not show the convergence of ILU(0) since it only takes 3 iterations to converge. This is exceptional because of the special near tridiagonal structure of A . Table 3 shows the number of nonzeros for each preconditioner. The wavelet based preconditioner requires much less amount of memory than SPAI does.

In Examples 2-4, the wavelet based preconditioner is most efficient in terms of convergence and storage. Although the convergence of the wavelet based preconditioner still depends on the mesh size (Figure 6), the dependence is less than ILU(0) and much less than SPAI. However, we would like to point out that this comparison is very rough since the preconditioners SPAI and ILU(0) take up much more nonzeros.

Besides rapid convergence, we can also see a tremendous gain in storage for the wavelet based preconditioner as n increases. This gain essentially comes from the wavelet compression. The larger n is, the more compression we can get. It is because the effect of singularity becomes less and less prominent as the singularity is only located along the diagonal.

Table 4 gives a comparison of the total operation counts for each method. The count estimate consists of the number of GMRES(20) iteration, the cost of matrix-vector multiply, application of the preconditioner and the number of inner products/saxpy operations. Since the number of inner products/saxpy operations depends on the iteration number, in the average, our operation counts estimate for one GMRES(20) iteration is:

$$\text{count} = \text{nnz}(A) + \text{nnz}(M) + 21n,$$

where $\text{nnz}(A)$ and $\text{nnz}(M)$ are the number of nonzeros of the matrix A and the preconditioner M respectively and n is the size of the matrix. The count for ILU(0) is normalized to one. The wavelet preconditioner shows a superior operation counts over all the other methods in Examples 2-4. In fact, the results is even better when n is larger. ILU(0) is exceptional good for the 1d problems in Example 1 as explained before. Despite that, the wavelet preconditioner still takes much smaller counts than the other two approximate inverses.

Finally Figure 7(a) shows that the wavelet based preconditioner does not always work. As mentioned before, we assume that the singularity of the Green's function is only at a point so that the wavelet transformed inverse has large entries near the main diagonal and our implementation can capture those successfully as shown in previous examples. However, for discontinuous coefficient problems, the Green's function has addition singularity along the discontinuities of the coefficients as shown in Fig 7(b). Hence the inverse is not as smooth as before. Thus our block diagonal structure may not completely capture the significant elements of the exact inverse. We should remark that the failure is mainly due to our current implementation. In principle, if we can locate the significant elements by some adaptive procedure (e.g. the one given in [21] and [11]), we should be able to obtain an effective approximate inverse preconditioner. However, such sophisticated adaptive searching technique is not fully developed yet for this class of problems and further investigation is needed.

Example	n	Wavelet SPAI	SPAI(0.4)	SPAI(0.2)	ILU(0)	No precondition.
1a	1024	32	>200	>200	3	>200
1b	1024	71	>200	>200	1	>200
2	1024	25	67	37	25	116
	4096	50	160	63	57	>200
3	1024	26	100	40	34	>200
	4096	66	>200	129	93	>200
4	1024	26	84	36	31	183
	4096	68	>200	126	89	>200
5	1024	>200	64	34	21	>200

TABLE 2
Number of GMRES(20) iterations

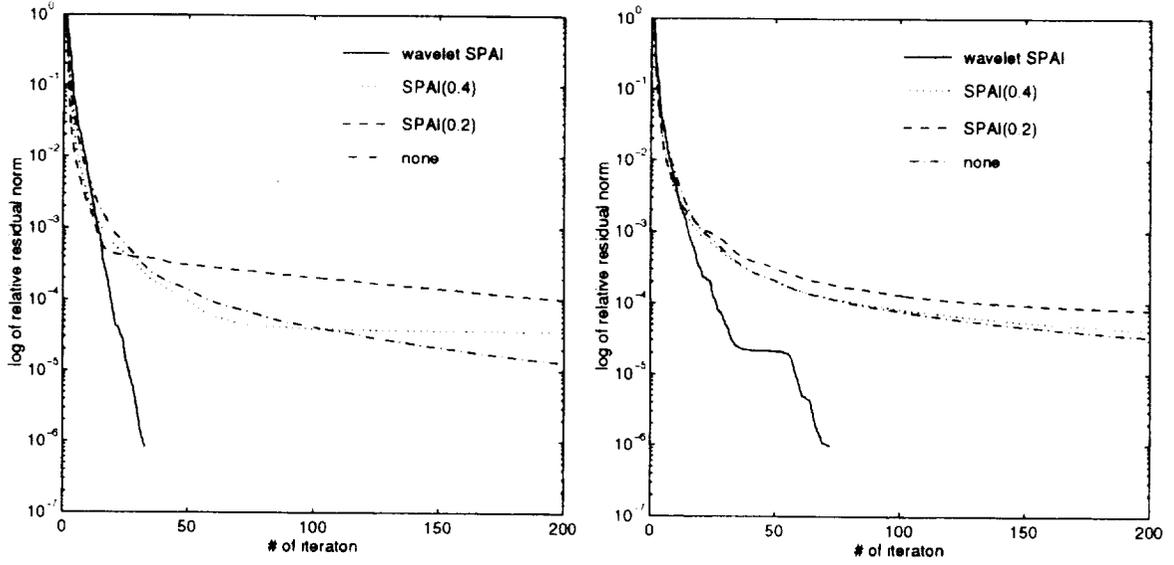


FIG. 2. Convergence behavior of GMRES(20) where (a) A =artificial matrix (b) A =1D Laplacian with Dirichlet and Neumann boundary conditions.

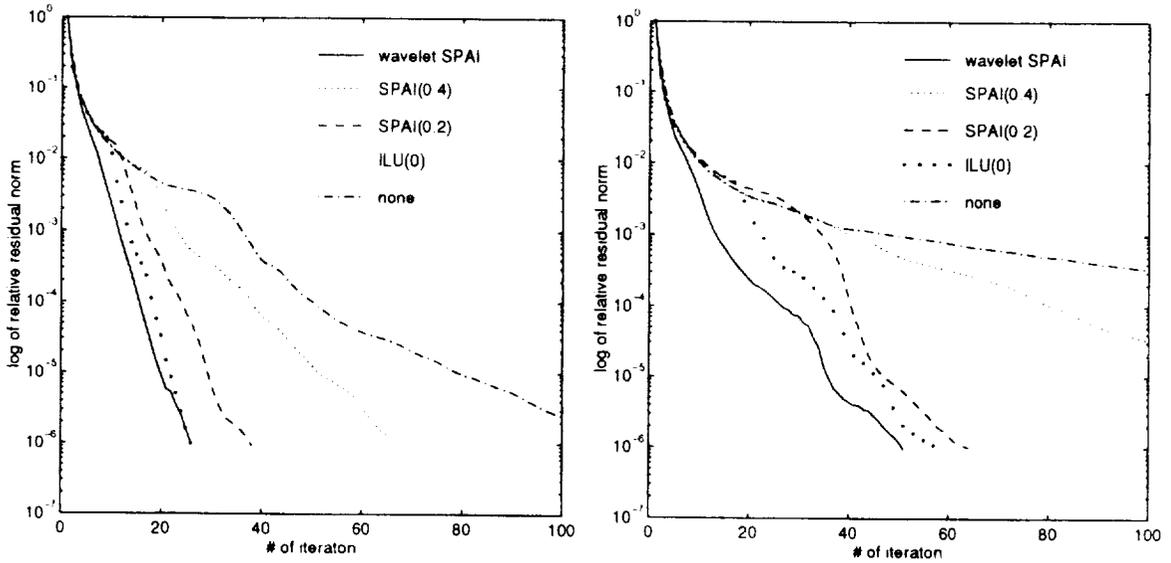


FIG. 3. Convergence behavior of GMRES(20) where A =2D Laplacian with size (a) $n=1024$ (b) $n=4096$.

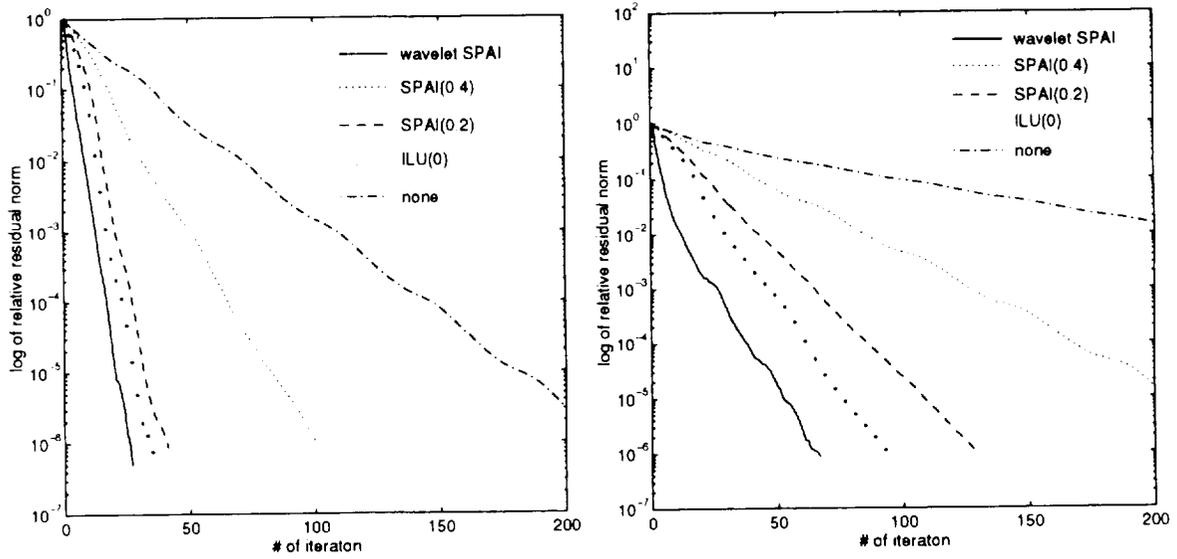


FIG. 4. Convergence behavior of GMRES(20) where A =variable coefficient operator with size (a) $n=1024$ (b) $n=4096$.

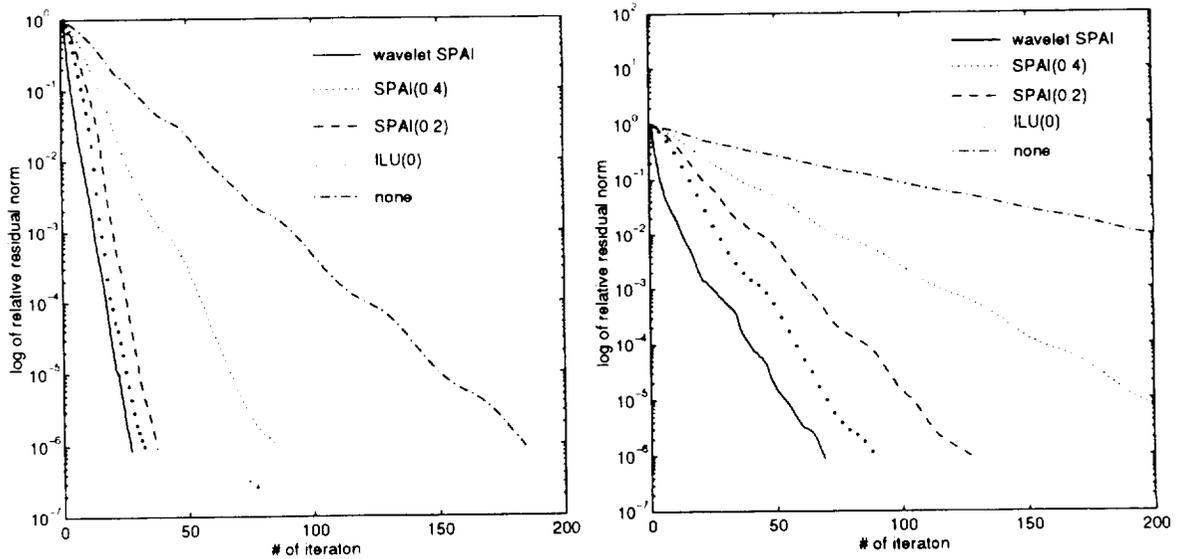


FIG. 5. Convergence behavior of GMRES(20) where A =Helical spring operator with size (a) $n=1024$ (b) $n=4096$.

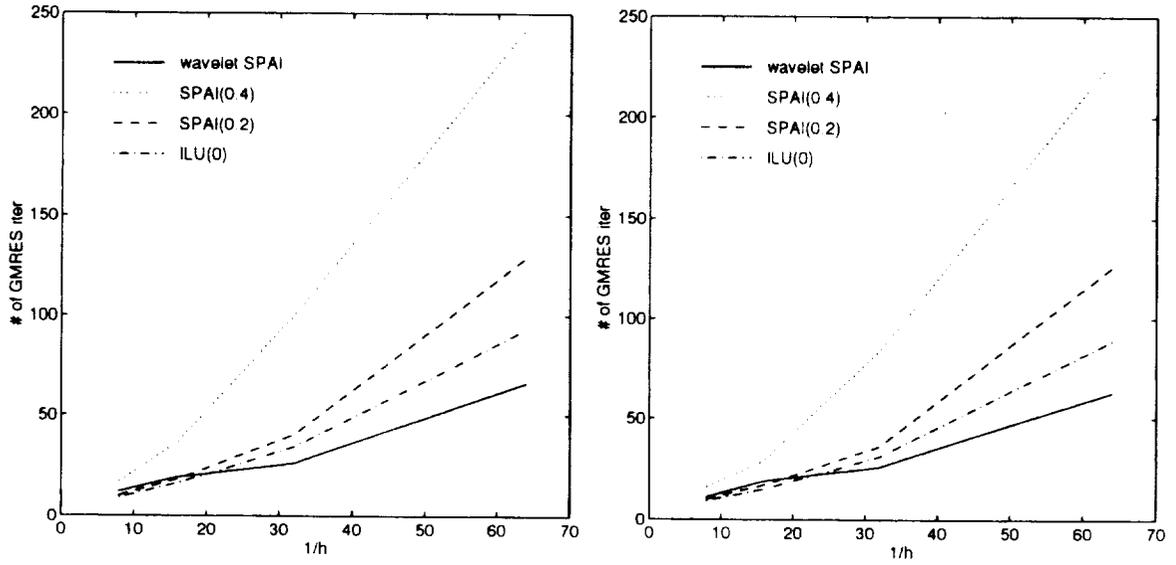


FIG. 6. Iteration number vs $1/h$, h =mesh size, for (a) variable coefficient (b) Helical spring.

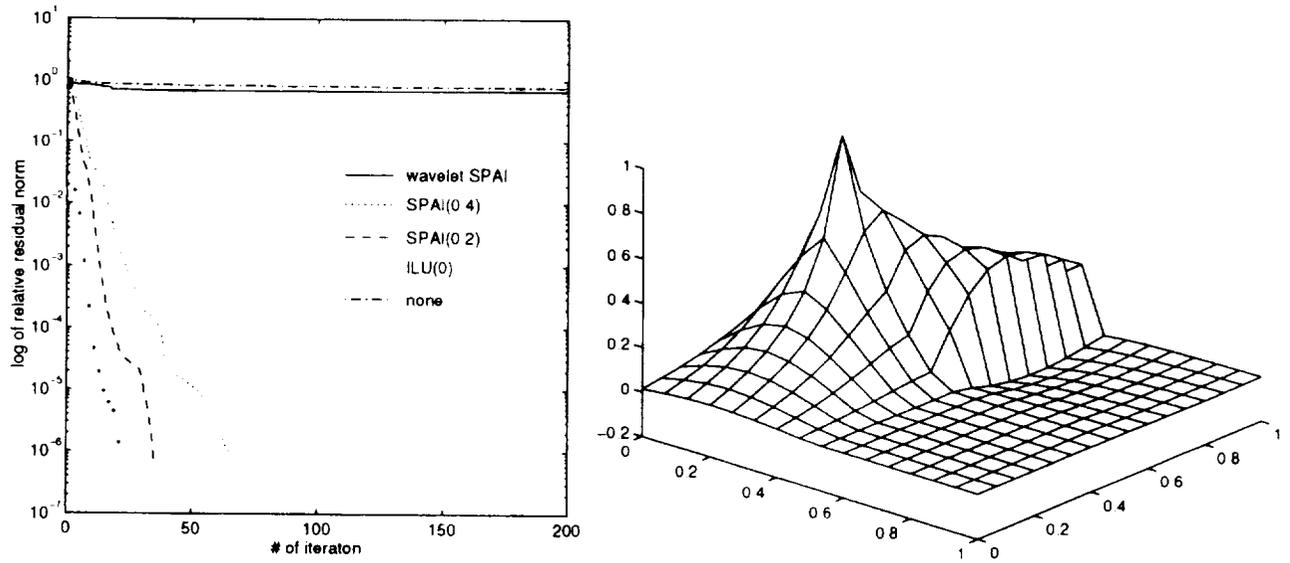


FIG. 7. (a) Convergence behavior of GMRES(20) where A =discont. coeffs. (b) The Green's function of A at the point $(0.1, 0.5)$

Example	n	Wavelet SPAI	SPAI(0.4)	SPAI(0.2)	ILU(0)
1a	1024	3544	5120	21504	3072
1b	1024	3544	5121	25425	3072
2	1024	3544	4624	16440	4992
	4096	6616	19472	69688	20224
3	1024	3544	4514	17260	4992
	4096	6616	18618	73936	20224
4	1024	3544	4624	16387	4992
	4096	6616	19472	69628	20224
5	1024	13464	5677	18952	4992

TABLE 3
Number of nonzero in approximate inverse

Example	n	Wavelet SPAI	SPAI(0.4)	SPAI(0.2)	ILU(0)
1a	1024	10	>72	>111	1
1b	1024	72	>215	>362	1
2	1024	0.95	2.65	2.02	1
	4096	0.78	2.79	1.54	1
3	1024	0.73	2.90	1.63	1
	4096	0.63	>2.12	1.98	1
4	1024	0.80	2.68	1.58	1
	4096	0.68	2.23	1.97	1
5	1024	>12	3.11	2.33	1

TABLE 4
Operation count estimate. The count for ILU(0) is normalized to 1.

7. Conclusion. Although our wavelet based preconditioner does not give rise to a "black box" solver, we have extended the potential applicability of approximate inverse to a larger class of problems, namely, matrices with piecewise smooth inverses. There are two main factors concerning our preconditioner: choice of basis and sparsity pattern. We have shown that for our block diagonal implementation, the wavelet basis is suitable for matrices with piecewise smooth inverse and singularity along the diagonal. Moreover, significant amount of storage can be saved. We should remark that other choices of basis are also feasible to solve specific problems, e.g. higher order wavelets, basis derived from multiresolution methods.

If the singularity of A^{-1} is along the diagonal, we have shown that block diagonal structure is sufficient. However, for more general situation, e.g. discontinuous coefficient case, where the singularity is not necessarily near the diagonal, more sophisticated adaptive searching procedure is needed to locate the sparsity pattern correctly.

Acknowledgment. We would like to thank Barry Smith for inspiring the idea of combining SPAI preconditioners with a wavelet basis and his valuable comments on this paper.

Appendix. We shall show that for 1D Dirichlet problem, the inverse of the discrete Laplacian operator A is actually a discrete Green's function. Consider the following problem,

$$\begin{aligned} -u''(x) &= f(x) & \forall x \in (0, 1) \\ u(0) &= y(1) = 0. \end{aligned}$$

Then the solution is given by $u(x) = \int_0^1 G(x, y)f(y)dy$, where $G(x, y)$ is a Green's function defined as

$$G(x, y) = \begin{cases} y(1-x) & 0 \leq y \leq x, \\ x(1-y) & x \leq y \leq 1. \end{cases}$$

Let a partition of $[0,1]$ be $\{x_0, x_1, \dots, x_{n+1}\}$. We want to evaluate y at the points $\{x_i\}$. By definition of u ,

$$u(x_i) = \int_0^1 G(x_i, y)f(y)dy.$$

We compute the integral by trapezoidal rule and we obtain

$$\begin{aligned} u(x_i) &\approx \frac{1}{n+1} \sum_{j=1}^n G(x_i, x_j)f(x_j) \\ &= \sum_{j=1}^n \frac{1}{n+1} G\left(\frac{i}{n+1}, \frac{j}{n+1}\right) f\left(\frac{j}{n+1}\right) \\ &= \sum_{j=1}^n G_{ij} f_j \end{aligned}$$

where

$$\begin{aligned} G_{ij} &= \frac{1}{n+1} G\left(\frac{i}{n+1}, \frac{j}{n+1}\right) \\ &= \begin{cases} \frac{1}{(n+1)^2} [j(n+1-i)] & j \leq i, \\ \frac{1}{(n+1)^2} [i(n+1-j)] & i \leq j. \end{cases} \end{aligned}$$

On the other hand, let A be the corresponding discrete Laplacian operator (second order central differencing) and $b = (b_j) = (f_j)$. Then the solution for $Ax = b$ is precisely $x = Gb$, where $G = (G_{ij})$ is defined above. Thus, rows of A^{-1} can be viewed as a discrete Green's function.

Results in higher dimensions can be also derived similarly.

REFERENCES

- [1] O. Axelsson. *Iterative Solution Methods*. Cambridge University Press, Cambridge, 1994.
- [2] M.W. Benson. Iterative solution of large scale linear systems. Master's thesis, Lakehead University, Thunder Bay, Ontario, 1973.
- [3] M.W. Benson and P.O. Frederickson. Iterative solution of large sparse linear systems arising in certain multidimensional approximation problems. *Utilitas Math.*, 22:127-140, 1982.
- [4] M. Benzi, C. D. Meyer, and M. Tuma. A sparse approximate inverse preconditioner for the conjugate gradient method. *SIAM J. Sci. Comput.*, 17, 1996.
- [5] M. Benzi and M. Tuma. A sparse approximate inverse preconditioner for nonsymmetric linear systems. Technical Report 653, Institute of Computer Science, Academy of Sciences of the Czech Republic, 1995.
- [6] G Beylkin, R. Coifman, and V. Rokhlin. Fast wavelet transforms and numerical algorithms I. *Comm. Pure Appl. Math.*, 44:141-184, 1991.
- [7] C. De Boor. Dichotomies for band matrices. *SIAM J. Numer. Anal.*, pages 894-907, 1980.
- [8] F.X. Canning and J.F. Scholl. Diagonal preconditioners for the EFIE using a wavelet basis. *IEEE Trans. on Antennas. and Propagation.*, Sept 1996. To appear.
- [9] T.F. Chan and T.P. Mathew. Domain decomposition algorithms. In *Acta Numerica*, pages 61-143. Cambridge University Press, 1994.
- [10] E. Chow and Y. Saad. Approximate inverse techniques for block-partitioned matrices. 1995. To appear in *SIAM J. Sci Comput.*
- [11] E. Chow and Y. Saad. Approximate inverse preconditioners for general sparse matrices. *Colorado Conference on Iterative Methods, April 5-9, 1994*. To appear in *SIAM J. Sci. Comput.*
- [12] J.D.F. Cosgrove, J.C. Diaz, and A. Griewank. Approximate inverse preconditionings for sparse linear systems. *Intern. J. Computer Math.*, pages 91-110, 1992.

- [13] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Comm. Pure Appl. Math.*, 41:909–996, 1988.
- [14] I. Daubechies. *Ten Lectures on Wavelets*, CBMS-NSF Series Appl. Math. SIAM, 1991.
- [15] S. Demko. Inverses of band matrices and local convergence of spline projections. *SIAM J. Numer. Anal.*, pages 616–619, 1977.
- [16] S. Demko, W.F. Moss, and P.W. Smith. Decay rates for inverses of band matrices. *Math. Comp.*, pages 491–499, 1984.
- [17] V. Eijkhout and B. Polman. Decay rates of inverses of banded m-matrices that are near to toeplitz matrices. *Linear Algebra and Its Appl.*, 109:247–277, 1988.
- [18] Bjorn Engquist, Stanley Osher, and Sifen Zhong. Fast wavelet based algorithms for linear evolution equations. *SIAM J. Sci. Comput.*, 15:755–775, 1994.
- [19] R. Glowinski, A. Rieder, R.O. Wells, Jr., and X. Zhou. A wavelet multigrid preconditioner for dirichlet boundary-value problems in general domains. Technical Report TR94-14, Rice Computational Mathematics Laboratory, 1994.
- [20] N. I. M. Gould and J. A. Scott. On approximate-inverse preconditioners. Technical Report RAL-TR-95-026, The Central Laboratory of the Research Councils, 1995.
- [21] M. Grote and T. Huckle. Parallel preconditioning with sparse approximate inverses. 1995. To appear in *SIAM J. Sci. Comput.*
- [22] M. Grote and H. Simon. Parallel preconditioning and approximate inverses on the connection machine. In *Scalable High Performance Computing Conference (SHPPC), 1992 Williamsburg, VA*, pages 76–83. IEEE Computer Science Press, 1992.
- [23] M. Grote and H. Simon. Parallel preconditioning and approximate inverses on the connection machine. In Richard Sincover et al, editor, *Sixth SIAM Conference on Parallel Processing for Scientific Computing II*, pages 519–523. SIAM, 1993.
- [24] W. Hackbusch. *Multi-grid Methods and Applications*. Springer-Verlag, Berlin, Heidelberg, 1985.
- [25] L.Yu. Kolotilina, A.A. Nikishin, and A.Yu. Yeregin. Factorized sparse approximate inverse (FSAI) preconditionings for solving 3D FE systems on massively parallel computers II. In R. Beauwens and P.de Groen, editors, *Iterative Methods in Linear Algebra, Proc of the IMACS International Symposium, Brussels, April 2-4, 1991*, pages 311–312, 1992.
- [26] L.Yu. Kolotilina and A.Yu. Yeregin. Factorized sparse approximate inverse preconditionings I. theory. *SIAM J. Matrix Anal. Appl.*, 14:45–58, 1993.
- [27] Ju.B. Lifshitz, A.A. Nikishin, and A.Yu. Yeregin. Sparse approximate inverse (FSAI) preconditionings for solving 3D CFD problems on massively parallel computers. In R. Beauwens and P.de Groen, editors, *Iterative Methods in Linear Algebra, Proc of the IMACS International Symposium, Brussels, April 2-4, 1991*, pages 83–84, 1992.
- [28] E. Ong. *Hierarchical Basis Preconditioners for Second Order Elliptic problems in Three Dimensions*. PhD thesis, University of Washington, Seattle, 1989.
- [29] B. Smith, P. Bjorstad, and W. Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, Cambridge, 1996.
- [30] W. P. Tang. Effective sparse approximate inverse preconditioners. In preparation, 1995.
- [31] P.S. Vassilevski and J. Wang. Stabilizing the hierarchical basis by approximate wavelets, I: Theory. Technical Report 95-47, Dept of Mathematics, UCLA, 1995.
- [32] P.S. Vassilevski and J. Wang. Stabilizing the hierarchical basis by approximate wavelets, II: Implementation and Numerical Results. Technical Report 95-48, Dept of Mathematics, UCLA, 1995.
- [33] H. Yserentant. On the multilevel splitting of finite element spaces. *Numer. Math.*, pages 379–412, 1986.

